DECLARATION

I, TAKAO OCHI, a Japanese Patent Attorney registered No. 10149, of Okabe International Patent Office at No. 602, Fuji Bldg., 2-3, Marunouchi 3-chome, Chiyoda-ku, Tokyo, Japan, hereby declare that I have a thorough knowledge of Japanese and English languages, and that the attached pages contain a correct translation into English of the priority document of Japanese Patent Application No. 11-085855 filed on March 29, 1999 in the name of CANON KABUSHIKI KAISHA.

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made, are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signed this 21st day of May, 2004

Takao Ochi

TAKAO OCHI

PATENT OFFICE
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: March 29, 1999

Application Number: Japanese Patent Application
No. 11-085855

Applicant(s): CANON KABUSHIKI KAISHA

April 21, 2000

Commissioner,
Japan Patent Office TAKAHIKO KONDO
(Seal)

Certificate No. 2000-3029107

085855/1999

[Name of Document] Application for Patent

[Reference No.] 3855004

[Date of Filing] March 29, 1999

[Addressee] Commissioner of the Patent Office

Mr. Takeshi ISAYAMA

[Int. Cl.] G06F 3/00

[Title of the Invention] PERIPHERAL APPARATUS, METHOD OF CONTROLLING THE PERIPHERAL APPARATUS, MEMORY MEDIUM, AND INFORMATION PROCESSING SYSTEM

[Number of Claims] 34

[Inventor]

[Address] c/o CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo

[Name] Masaki NISHIYAMA

[Applicant for Patent]

[Id. No.] 000001007

[Address] 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo

[Name] CANON KABUSHIKI KAISHA

[Representative] Fujio MITARAI

[Phone No.] 03-3758-2111

[Agent]

[Id. No.] 100069877

[Address] c/o CANON KABUSHIKI KAISHA, 30-2,

Shimomaruko 3-chome, Ohta-ku, Tokyo

[Patent Attorney]

[Name] Giichi MARUSHIMA

[Phone No.] 03-3758-2111

[Application Fees]

[Prepayment Registration No.] 011224

[Amount of Payment] 21000

[List of Documents Attached]

[Name of Document] Specification 1

[Name of Document] Drawings 1

[Name of Document] Abstract 1

[No. of General Power of Attorney] 9703271

[Proof] Required

[Name of Document] SPECIFICATION

[Title of the Invention] PERIPHERAL APPARATUS, METHOD OF CONTROLLING THE PERIPHERAL APPARATUS, MEMORY MEDIUM, AND INFORMATION PROCESSING SYSTEM

[Claims]

[Claim 1] A peripheral apparatus comprising:

connecting means for connecting to a host computer;

first means for, in response to a data reading request issued from the host computer, notifying said connecting means of response data when the response data is ready and notifying said connecting means that a response cannot be made when the response data is not ready;

second means for notifying said connecting means of the absence of data to be responded to in response to the data reading request issued from the host computer; and

switching means for switching said first means and said second means in accordance with a status of the peripheral apparatus.

[Claim 2] A peripheral apparatus according to Claim 1, further comprising second connecting means that is different from said connecting means,

wherein said switching means switches to said second means when the data supplied from said second connecting means is processed.

[Claim 3] A peripheral apparatus according to Claim 1,

wherein said connecting means conforms to a USB standard or an IEEE1394 standard.

[Claim 4] A peripheral apparatus according to Claim 2, wherein said connecting means conforms to a USB standard or an IEEE1394 standard and said second connecting means conforms to an IEEE1284 standard.

[Claim 5] A peripheral apparatus according to Claim 1 or 2, wherein said first means notifies said connecting means of the response data by using a packet and notifies said connecting means of the fact that the response cannot be made by using a Nak packet, and said second means notifies said connecting means of the absence of the data to be responded to by using a blank packet.

[Claim 6] A peripheral apparatus according to Claim 1, wherein said peripheral apparatus includes a printer.

[Claim 7] A peripheral apparatus according to Claim 1, wherein said peripheral apparatus includes a scanner.

[Claim 8] A peripheral apparatus according to Claim 1, wherein said peripheral apparatus includes a facsimile.

[Claim 9] A peripheral apparatus according to Claim 6, 7, or 8, wherein said switching means switches to said second means at a timing when a printer engine or a scanner engine operates.

[Claim 10] A peripheral apparatus according to Claim 6 or 8, wherein said switching means switches to said second

means at a timing when print data is received or at a timing when expansion of the print data is finished.

[Claim 11] A method of controlling a peripheral apparatus comprising connecting means for connecting to a host computer; first means for, in response to a data reading request issued from the host computer, notifying said connecting means of response data when the response data is ready and notifying said connecting means that a response cannot be made when the response data is not ready; and second means for notifying said connecting means of the absence of data to be responded to in response to the data reading request issued from the host computer,

wherein the method includes the step of switching said first means and said second means in accordance with a status of the peripheral apparatus.

[Claim 12] A method of controlling a peripheral apparatus according to Claim 11, wherein said peripheral apparatus further includes second connecting means that is different from said connecting means, and wherein said switching step switches to said second means when the data supplied from said second connecting means is processed.

[Claim 13] A method of controlling a peripheral apparatus according to Claim 11, wherein said connecting means conforms to a USB standard or an IEEE1394 standard.

[Claim 14] A method of controlling a peripheral apparatus

according to Claim 12, wherein said connecting means conforms to a USB standard or an IEEE1394 standard and said second connecting means conforms to an IEEE1284 standard.

[Claim 15] A method of controlling a peripheral apparatus according to Claim 11 or 12, wherein said first means notifies said connecting means of the response data by using a packet and notifies said connecting means of the fact that the response cannot be made by using a Nak packet, and said second means notifies said connecting means of the absence of the data to be responded to by using a blank packet.

[Claim 16] A method of controlling a peripheral apparatus according to Claim 11, wherein said peripheral apparatus includes a printer.

[Claim 17] A method of controlling a peripheral apparatus according to Claim 11, wherein said peripheral apparatus includes a scanner.

[Claim 18] A method of controlling a peripheral apparatus according to Claim 11, wherein said peripheral apparatus includes a facsimile.

[Claim 19] A method of controlling a peripheral apparatus according to Claim 16, 17, or 18, wherein said switching step switches to said second means at a timing when a printer engine or a scanner engine operates.

[Claim 20] A method of controlling a peripheral apparatus according to Claim 16 or 18, wherein said switching means

switches to said second means at a timing when print data is received or at a timing when expansion of the print data is finished.

[Claim 21] A memory medium for storing a control program of a peripheral apparatus comprising connecting means for connecting to a host computer; first means for, in response to a data reading request issued from the host computer, notifying said connecting means of response data when the response data is ready and notifying said connecting means that a response cannot be made when the response data is not ready; and second means for notifying said connecting means of the absence of data to be responded to in response to the data reading request issued from the host computer,

wherein the control program includes the step of switching said first means and said second means in accordance with a status of the peripheral apparatus.

[Claim 22] A memory medium according to Claim 21, wherein said peripheral apparatus further includes second connecting means that is different from said connecting means, and wherein said switching step switches to said second means when the data supplied from said second connecting means is processed.

[Claim 23] A memory medium according to Claim 21, wherein said connecting means conforms to a USB standard or an IEEE1394 standard.

[Claim 24] A memory medium according to Claim 22, wherein said connecting means conforms to a USB standard or an IEEE1394 standard and said second connecting means conforms to an IEEE1284 standard.

[Claim 25] A memory medium according to Claim 21 or 22, wherein said first means notifies said connecting means of the response data by using a packet and notifies said connecting means of the fact that the response cannot be made by using a Nak packet, and said second means notifies said connecting means of the absence of the data to be responded to by using a blank packet.

[Claim 26] A memory medium according to Claim 21, wherein said peripheral apparatus includes a printer.

[Claim 27] A memory medium according to Claim 21, wherein said peripheral apparatus includes a scanner.

[Claim 28] A memory medium according to Claim 21, wherein said peripheral apparatus includes a facsimile.

[Claim 29] A memory medium according to Claim 26, 27, or 28, wherein said switching step switches to said second means at a timing when a printer engine or a scanner engine operates.

[Claim 30] A memory medium according to Claim 26 or 28, wherein said switching means switches to said second means at a timing when print data is received or at a timing when expansion of the print data is finished.

[Claim 31] An information processing system comprising a host computer and the peripheral apparatus according to Claim 1, the host computer including issuance means for issuing a data reading request to said peripheral apparatus and processing means for causing said issuance means to issue the data reading request in response to a notification from said peripheral apparatus that a response cannot be made and for performing a corresponding process in response to a notification from the peripheral apparatus that no response data or no data to be responded to exists.

[Claim 32] A peripheral apparatus according to Claim 1, wherein the host computer includes issuance means for issuing the data reading request to said peripheral apparatus and processing means for causing said issuance means to issue the data reading request in response to a notification from said peripheral apparatus that the response cannot be made and for performing a corresponding process in response to a notification from the peripheral apparatus that no response data or no data to be responded to exists.

[Claim 33] A method of controlling a peripheral apparatus according to Claim 11, wherein the host computer includes issuance means for issuing the data reading request to said peripheral apparatus and processing means for causing said issuance means to issue the data reading request in response

to a notification from said peripheral apparatus that the response cannot be made and for performing a corresponding process in response to a notification from the peripheral apparatus that no response data or no data to be responded to exists.

[Claim 34] A memory medium according to Claim 21, wherein the host computer includes issuance means for issuing the data reading request to said peripheral apparatus and processing means for causing said issuance means to issue the data reading request in response to a notification from said peripheral apparatus that the response cannot be made and for performing a corresponding process in response to a notification from the peripheral apparatus that no response data or no data to be responded to exists.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention]

The invention relates to a peripheral apparatus connected to an information processing apparatus, a control method for the peripheral apparatus, a memory medium, and an information processing system.

[0002]

[Description of the Related Art]

Hitherto, when a host receives data from a printer, for example, according to the Universal Serial Bus (USB)

standard, a Bulk-In request is issued. Generally, according to such a standard, in response to the request, the printer returns data or, if the data is not ready, the printer returns a Nak.

[0003]

[Problems to be Solved by the Invention]

However, in conventional techniques, for example, an inexpensive printer cannot promptly return the data during printing or the like because of a limitation of hardware in that a CPU for controlling the USB communication and a CPU for print control are identical. Consequently, there are big problems in that a fairly large number of Bulk-In and Nak responses are mixed on the USB bus to cause a waste and the performance of a host PC remarkably deteriorates due to the issuance of the Bulk-In.

[0004]

In a printer having a dual I/F confirming, for example, to the USB standard and an IEEE1284 standard, such problems become more remarkable during the printing according to the IEEE1284.

[0005]

It is, therefore, an object of the invention to provide a peripheral apparatus, a control method for the peripheral apparatus, and a memory medium, in which the performance of a connected information processing apparatus (host computer)

is not deteriorated.

[0006]

[Means for Solving the Problems]

To accomplish the above object, a peripheral apparatus according to the invention includes connecting means for connecting to a host computer; first means for, in response to a data reading request issued from the host computer, notifying the connecting means of response data when the response data is ready and notifying the connecting means that a response cannot be made when the response data is not ready; second means for notifying the connecting means of the absence of data to be responded to in response to the data reading request issued from the host computer; and switching means for switching the first means and the second means in accordance with a status of the peripheral apparatus.

[0007]

The present invention provides a method of controlling a peripheral apparatus. The peripheral apparatus includes connecting means for connecting to a host computer; first means for, in response to a data reading request issued from the host computer, notifying the connecting means of response data when the response data is ready and notifying the connecting means that a response cannot be made when the response data is not ready; and second means for notifying

the connecting means of the absence of data to be responded to in response to the data reading request issued from the host computer. The method includes the step of switching the first means and the second means in accordance with a status of the peripheral apparatus.

[0008]

The present invention provides a memory medium for storing a control program of a peripheral apparatus. The peripheral apparatus includes connecting means for connecting to a host computer; first means for, in response to a data reading request issued from the host computer, notifying the connecting means of response data when the response data is ready and notifying the connecting means that a response cannot be made when the response data is not ready; and second means for notifying the connecting means of the absence of data to be responded to in response to the data reading request issued from the host computer. The control program includes the step of switching the first means and the second means in accordance with a status of the peripheral apparatus.

[0009]

[Description of the Embodiments]

A printing apparatus according to an embodiment of the invention will be described in detail below with reference to the attached drawings. Although the USB is described as

communicating means for convenience of explanation, it will be easily understood that the invention can also be embodied by other communicating means for performing packet communication. Although a printer is described as a peripheral apparatus, the invention can also be embodied by a peripheral apparatus such as a scanner or a facsimile.

[0010]

(First Embodiment)

Embodiments of the invention will now be described in detail with reference to the attached drawings.

[0011]

In the embodiments, Control, Bulk-In, and Bulk-Out pipes are used in conformity with USB Printer Spec. 1.0. (Refer to Fig. 2 and to USB standard 1.00 for the definition of those pipes.) The Control pipe is mainly used for control such as printer reset. Print data is transmitted to the printer by using Bulk-Out pipe. The Bulk-In pipe is used to return a status or the like of the printer to the host.

[0012]

Fig. 1 is a diagram showing a structure according to a first embodiment of the present invention. Reference numeral 50 denotes a host computer; reference numeral 60 denotes a printer; reference numeral 100 denotes a central processing unit (CPU) for performing various processes;

reference numeral 200 denotes a random access memory (RAM); reference numeral 300 denotes a read only memory (ROM) for storing a control program regarding a flowchart of the present invention, which will be described below, and a control program for the printer; reference numeral 400 denotes a printer engine; reference numeral 500 denotes a Bulk-Out FIFO for receiving the print data to be supplied to the printer engine 400; reference numeral 600 denotes a Bulk-In FIFO for the Bulk-In pipe, which is used for data to be returned to the host, shown in Fig. 9, such as status information of the printer; reference numeral 700 denotes a serial interface engine (SIE) of the USB; and reference numeral 800 denotes a USB cable. In conventional techniques, the Bulk-In FIFO 600 is directly connected to the SIE 700. According to the present invention, reference numeral 900 denotes a blank-packet generator for generating a blank packet shown in Fig. 10, reference numeral 1000 denotes a switch for switching the Bulk-In FIFO 600 to the blank-packet generator 900 in accordance with an instruction from the CPU 100, and reference numeral 10 denotes a system bus.

[0013]

Control of the first embodiment in the above structure will be described below with reference to the drawings.

[0014]

First, an outline of a conventional technique will be

described. The print data supplied from the host is encoded by a USB host controller (not shown) of a host machine and flows through the USB cable 800 on a USB bus as Bulk-Out data. The data is decoded in the SIE 700 and stored in the Bulk-Out FIFO 500. The CPU 100 acquires the data stored in the FIFO 500 and transfers the acquired data to the printer engine 4 in order to perform printing. The print data is transmitted in this manner.

[0015]

A flow of the status information of the printer will now be described. Generally, a printer driver not only transmits the print data to the printer, but also provides a printer status, such as paper jam or absence of paper, to a user. Such a function is controlled by a program called a status monitor. The status monitor periodically issues a status request to the printer in order to obtain the status. The status request issued in the above manner is sent to the USB host controller via many application program interfaces (APIs), which are provided by the operating system (OS), and drivers (refer to Fig. 3). The status request serves as a Bulk-In request in the USB host controller and is issued onto the USB bus. The SIE 700 decodes the Bulk-In request and notifies the CPU 100 of the decoded request. The CPU 100, which detects the Bulk-In request, stores the status of the printer engine 400 in the Bulk-In FIFO 600. Next, the

status data stored in the FIFO 600 is encoded by the SIE 700 and is returned to the host through the USB cable 800. This returned data is decoded by the USB host controller and is sent to the status monitor via the API and the drivers in an order opposite to the order during the request. Finally, the status monitor also notifies the operator of the acquired data as a printer status by using the API for display.

[0016]

Since the CPU 100 simultaneously performs the control of the printer engine and the process of the USB interface in an inexpensive printer, there is a case where a response cannot be timely made in response to the Bulk-In request due to a problem on processing performance. In such a case, since there is no data in the Bulk-In FIFO 600, the SIE 700 returns a Nak (a packet indicating that a response cannot be made because of a busy status) shown in Fig. 11 to the host according to the USB standard. The USB host controller, which receives the Nak, automatically issues the Bulk-In request onto the USB bus again. The SIE 700 returns the Nak again in response to the issued Bulk-In and the USB host controller issues the Bulk-In request again in response to the returned Nak. Such endless repetition wastefully occurs. Furthermore, during this repetition, since the Nak is automatically processed in the host controller and the

control is not returned to the status monitor, the operator cannot be notified of the status. It is also reported that there is a case where the performance of the entire system is deteriorated by up to about 50% in the USB host controller due to an overhead required for the issue of the Bulk-In request.

[0017]

The control of the printer of the present invention will now be described in detail. According to the present invention, the Bulk-In FIFO 600 is not directly connected to the SIE but is connected to the SIE through the switch 1000. The switch 1000, which is also connected to the blank-packet generator 900, controls the connection to the SIE 700 in accordance with an instruction from the CPU 100. With such a structure, when the print data is transmitted from the host PC, the data is first stored in the Bulk-Out FIFO 500 by the conventional technique. The CPU 100 operates the switch 1000 in advance, before controlling the printer engine in order to print this data, to connect the blank-packet generator 900 to the SIE 700. In the first embodiment, the switching operation is performed by writing "01h" into an I/O port 80h (a small letter "h" represents the hexadecimal notation) of the CPU 100. When the control of the printer engine is finished, the CPU 100 operates the switch 1000 again (by writing "00h" into the I/O port 80h)

to connect the Bulk-In FIFO 600 to the SIE 700.

[0018]

Even when the CPU 100 receives the Bulk-In request from the USB host controller during the control of the printer engine, it is possible to automatically return a blank packet in response to the request because the blank-packet generator is connected to the SIE 700. The USB host controller, which receives the blank packet, returns a signal indicating the absence of reception data (NumofBytesRead = 0) to a driver layer 53. Furthermore, an API layer 52 also returns the signal indicating the absence of reception data to a status monitor 51 in the same manner. Since the control is returned to the status monitor, unlike conventional techniques, the status monitor can notify the operator of the absence of the reception data (usually, the busy status of the printer).

[0019]

According to the invention, when the USB host controller receives the blank packet from the SIE 700, the Bulk-In transaction is finished, thus avoiding the degradation of the performance.

[0020]

Fig. 12 is a flowchart showing the print control process according to the first embodiment. This flow is executed at a timing when the print data is received and

expanded and the printer engine is activated to start printing. First, in step S51, "01" is written into the I/O port 80h. In step S52, the printer engine is controlled. However, the blank packet is returned by the writing in step S51 in response to the Bulk-In request under control. When the control of the printer engine is finished, in step S53, "00" is finally written into the I/O port 80h and the process is finished.

[0021]

This print control process may be executed at a timing when the print data is received from the host or at a timing when the expansion of the print data is finished.

[0022]

As described above, according to the present invention, not only the operator can be timely notified of the printer status, but also the degradation of the performance of the host PC can be avoided.

[0023]

(Second Embodiment)

Fig. 4 is a diagram showing a structure according to a second embodiment. A Centronics interface (IEEE1284) is further added to the structure of the first embodiment, thereby enabling the printer to be shared by two host PCs. With this structure, more significant effects of the invention can be realized. Generally, in the case of double

interface as mentioned above, the interface used first has the priority and the other interface can be used after the end of a print job. In other words, with this structure, a period during which the printer cannot respond to the Bulk-In request from the host PC is very long (two to three minutes or longer). (This is because, in conventional techniques, it is necessary for the CPU 100 not only to control the printer engine control but also to process the Centronics I/F and, therefore, the printer cannot respond to the Bulk-In request through the USB I/F until the end of the printing.) Specifically, now assuming that the operator of a PC1 has started the printing while a PC2 was using the Centronics I/F, although the status request from the status monitor is issued as Bulk-In onto the USB bus, the Nak is returned in response to this request until the print job in the PC2 is finished (usually, for two to three minutes), thus remarkably deteriorating the performance of the host PC1. Since the control is not returned to the status monitor as described above in first the embodiment, the operator of the host PC1 is not notified of the busy status of the printer during this time period.

[0024]

According to the present invention, it is easily understood that it is sufficient to operate the switch 1000 in order to connect the blank-packet generator to the SIE

700 before starting the printing through the Centronics I/F. Thus, even in the printing through the Centronics I/F, the blank packet can be timely returned, the performance of the host PC1 is not deteriorated, and the operator can be notified of the busy status of the printer. After the printing through the Centronics I/F is finished, the switch 1000 is switched to the Bulk-In FIFO 600 in preparation for the subsequent print job. On the contrary, while the host PC1 is using the USB I/F, the host PC2 is notified of the busy status of the printer by setting the busy status of the Centronics I/F of the printer to the high level by a conventional technique (refer to Figs. 4 and 5).

[0025]

Fig. 13 is a flowchart showing the print control process according to the second embodiment. First, in step S61, it is determined whether data exists in the Bulk-Out FIFO for the USB. If the determination is "true", the process proceeds to step S66. If the determination is "false", the process proceeds to step S62 and determines whether Centronics data exists. If the determination is "false", the process returns to step S61. If the determination is true in step S62, in step S63, "01" is written into the I/O port 80h. In step S64, the printing process of the Centronics data is completed. In step S65, "00h" is written into the I/O port 80h and the process

returns to step S61. If the determination is "true" in step S61, in step S66, the Centronics I/F is set to a busy status (#11pin - high). In step S67, the print job through the USB I/F is completed. In step S68, the Centronics I/F is set to a ready status (#11 - low) and the process returns to step S61. Since the blank packet is returned in response to the Bulk-In request in step S63 even when the print job is supplied through the USB I/F while printing the Centronics data, no Nak is generated, thereby making it possible to notify the operator of the fact that the printer is busy. Similarly, since the Centronics I/F is set to busy status in step S66 while performing the printing through the USB I/F, the operator can also be notified of the busy status. (Although time-out is used as usual means for determining the busy status of the Centronics I/F, a detailed description is omitted here because the time-out is a well-known technique.)

[0026]

(Third Embodiment)

Fig. 6 is a diagram showing a structure according to a second embodiment. The structure differs from that of the first embodiment in that the printer is provided with a HUB function and an additional printer can be attached to the downstream.

[0027]

(Fourth Embodiment)

According to a fourth embodiment, cables that are connected to the upstream and the downstream in the structure of first the embodiment can be connected or disconnected in a current-supplying state. Since the structure of the fourth embodiment is substantially similar to that of the first embodiment, a detailed description is omitted here. The fourth embodiment will now be described in detail with reference to the drawings.

[0028]

Fig. 7 is a diagram showing signal lines included in a communication apparatus of the invention and their drivers.

[0029]

A shield twisted cable 401 (hereinafter referred to as a cable 401) including signal lines data1 and data2 connects a PC-side hub (repeater) 402 to a device side 403 (hub or node). In detail, each signal line is connected to a transceiver 404 to electrically exchange data with the transceiver 404. Resistors R1 and R2, which are connected to the corresponding signal lines, prevent the signal lines from being set to a high impedance.

[0030]

Each of the transceivers 404 and 404', which contains a differential-amplifier-type input/output device, a port for reading a voltage of each signal line, a serial-parallel

converter, and the like, controls electric signals in the signal lines data1 and data2. The signal lines data1 and data2 can serially transmit a control signal of a PC 102 and a signal supplied from a node 104 in accordance with a predetermined protocol.

[0031]

According to the present invention, when no signal is transmitted and received, it is indicated that a device is connected to one end (the node side) of the network if the signal line data1 is set to the high level and the signal line data2 is set to the low level, and that no device is connected if the signal line data1 is set to the low level and the signal line data2 is set to the low level.

[0032]

At the device side 403, the signal line data1 is connected to an R-controller 405 through a resistor R3.

[0033]

Fig. 8 includes diagrams showing timings at which the cable 401 is connected and disconnected and voltage changes of the signal line data1.

[0034]

Reference letters VOl and VOh in Fig. 8 indicate low and high detectable voltages of a port connected to the signal line data1 at the host side 402, respectively.

[0035]

Fig. 8-1 is a diagram showing a connecting sequence.

[0036]

Reference numeral 501 denotes a time point when the cable 401 has been connected. In this instance, the R-controller 405 connected to the resistor R3 generates a voltage of 5V and the voltage of the signal line data1 rises in accordance with a wiring capacitance of the resistor R3 and the cable 401. The voltage of the signal line data1 exceeds VOh after the elapse of a certain time T1 (time point 502) and the port input is recognized to be at the high level. Hence, it is possible to detect that a device has been connected to the downstream port.

[0037]

Fig. 8-2 is a diagram showing a disconnecting sequence of the cable 401.

[0038]

Reference numeral 503 denotes a time point when the cable 401 has been disconnected. The voltage of the signal line data1 drops in accordance with a wiring capacitance of the resistor R1 and the signal line data1. The voltage of the signal line data1 exceeds VOl after the elapse of a certain time T2 (time point 504) and the port input is recognized to be at the low level. Hence, it is possible to detect that the device has been disconnected from the downstream port.

[0039]

(Other Embodiments)

The invention is not restricted by the kind of a network and can be applied to networks having various interfaces. For example, the invention can be applied to interfaces called USB or IEEE1394. It will be also obviously understood by those skilled in the art that the invention can be similarly applied not only to printers but also to other information apparatuses such as facsimile apparatuses having a printer engine or scanners where the printer engine is replaced with a scanner engine.

[0040]

In peripheral apparatuses that can notify the information processing apparatus of the busy status (Nak packet), provision of means for indicating the absence of the communication information (packet without response data) other than the busy status and of means capable of exclusively selecting the busy status and the absence of the communication information can have a large effect of eliminating waste requests and the busy status on the communication bus. Furthermore, the operator can be timely notified of the status of the information processing apparatus.

[0041]

[Advantages]

As described above, according to the present invention, it is possible to provide a peripheral apparatus, a method of controlling the peripheral apparatus, and a memory medium which do not deteriorate the performance of the connected information processing apparatus (host computer).

[Brief Description of the Drawings]

[Fig. 1]

Fig. 1 is a diagram showing a structure according to a first embodiment.

[Fig. 2]

Fig. 2 is a diagram showing the structure of a USB endpoint of the first embodiment.

[Fig. 3]

Fig. 3 is a diagram showing a software configuration of a host PC and a printer.

[Fig. 4]

Fig. 4 is a diagram showing a structure according to a second embodiment.

[Fig. 5]

Fig. 5 is a diagram showing signal lines according to an IEEE1284 standard.

[Fig. 6]

Fig. 6 is a diagram showing a structure according to a third embodiment.

[Fig. 7]

Fig. 7 is a diagram showing communication signal lines and their drivers according to a fourth embodiment.

[Fig. 8]

Fig. 8 includes diagrams showing timings at which a cable 401 is connected and disconnected and voltage changes of a signal line data1.

[Fig. 9]

Fig. 9 is a diagram showing a data packet.

[Fig. 10]

Fig. 10 is a diagram showing a blank packet.

[Fig. 11]

Fig. 11 is a diagram showing a Nak packet.

[Fig. 12]

Fig. 12 is a flowchart showing a print control process of the first embodiment.

[Fig. 13]

Fig. 13 is a flowchart showing a print control process of the second embodiment.

[Reference Numerals]

VOL: low detectable voltage

VOH: high detectable voltage

VSS: ground level

[Fig. 1]

50 HOST

a CONVENTIONAL TECHNIQUE

900 BLANK-PACKET GENERATOR

400 PRINTER ENGINE

60 PRINTER

[Fig. 2]

[Fig. 3]

50 HOST

51 STATUS MONITOR

52 API LAYER

53 DRIVER LAYER

54 HOST CONTROLLER

60 PRINTER

55 PRINTING MECHANISM

56 FIRMWARE

57 SIE (Serial I/F Engine)

[Fig. 4]

[Fig. 5]

[Fig. 6]

50 HOST

51 STATUS MONITOR

52 API LAYER

53 DRIVER LAYER

54 HOST CONTROLLER

60 PRINTER

55 PRINTING MECHANISM

56 FIRMWARE

57 SIE (Serial I/F Engine)

[Fig. 7]

b PC-SIDE HUB (REPEATER)

404 TRANSCEIVER

401 SHIELD TWISTED CABLE

c DEVICE SIDE (NODE OR HUB)

405 R-CONTROLLER

404 TRANSCEIVER

[Fig. 8]

Fig. 8-1

d CONNECTION

e DETECT CONNECTION

Fig. 8-2

f DISCONNECTION

g DETECT DISCONNECTION

[Fig. 9]

[Fig. 10]

[Fig. 11]

[Fig. 12]

h PRINTING PROCESS

S52 CONTROL PRINTER ENGINE

i END

[Fig. 13]

j PRINTING PROCESS

S61 DOES DATA EXIST IN USB?

S62 DOES CENTRONICS DATA EXIST?

S64 PRINTING PROCESS OF CENTRONICS DATA

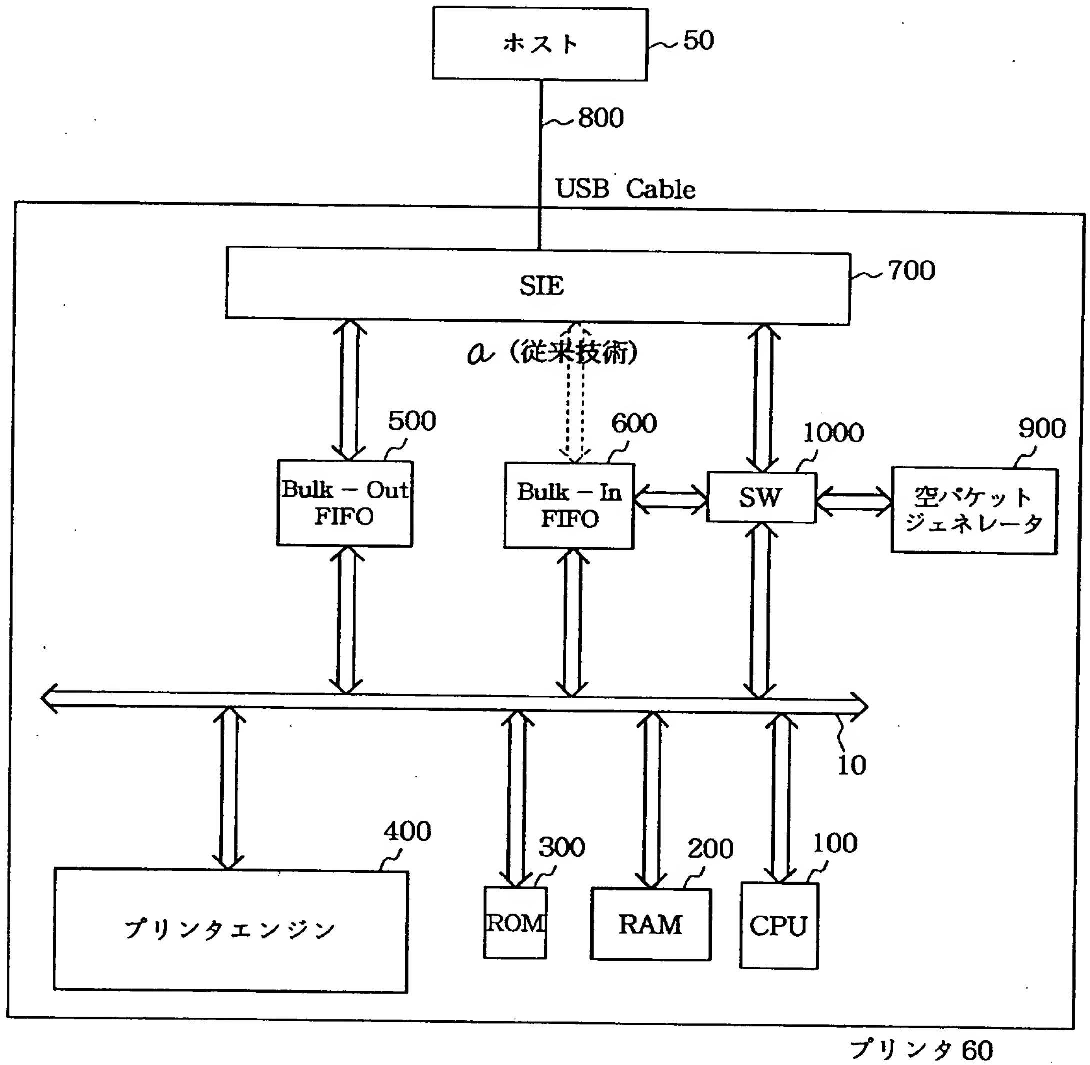
S67 PRINTING PROCESS OF USB I/F

k YES

l NO

【書類名】　図面

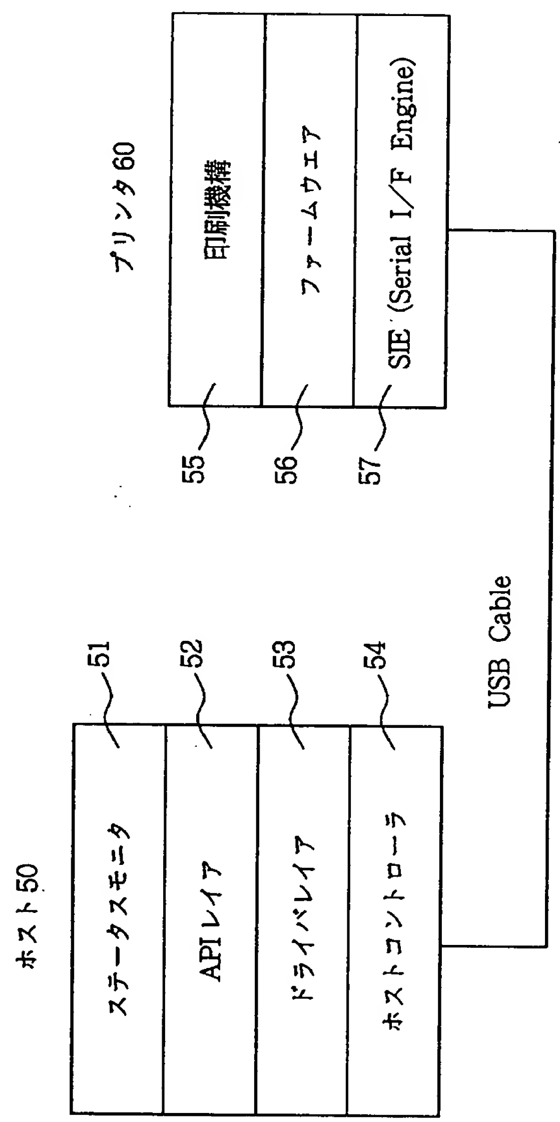
【図１】


ホスト
50
800
USB Cable
SIE
700
a（従来技術）
500
600
1000
900
Bulk－Out FIFO
Bulk－In FIFO
SW
空パケット ジェネレータ
10
400
300
200
100
プリンタエンジン
ROM
RAM
CPU
プリンタ60

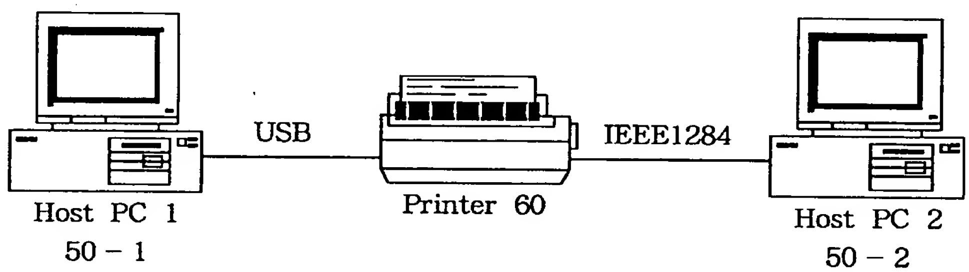
【図２】

EndPoint 0	Control
EndPoint 1	Bulk－Out
EndPoint 2	bulk－In

【図３】


プリンタ60
印刷機構
ファームウエア
SIE (Serial I/F Engine)
55
56
57
USB Cable
ホスト50
ステータスモニタ
APIレイア
ドライバレイア
ホストコントローラ
51
52
53
54

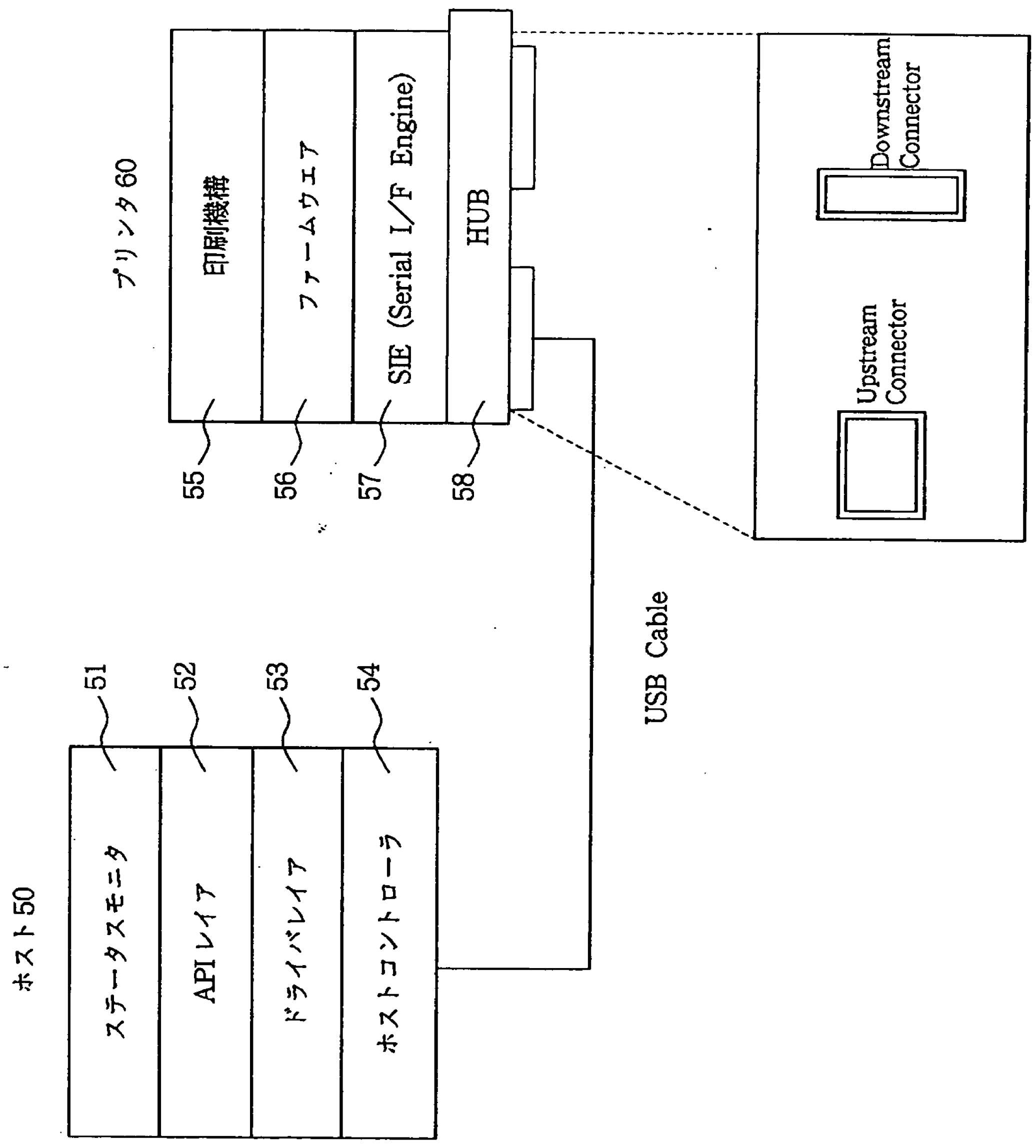
【図4】



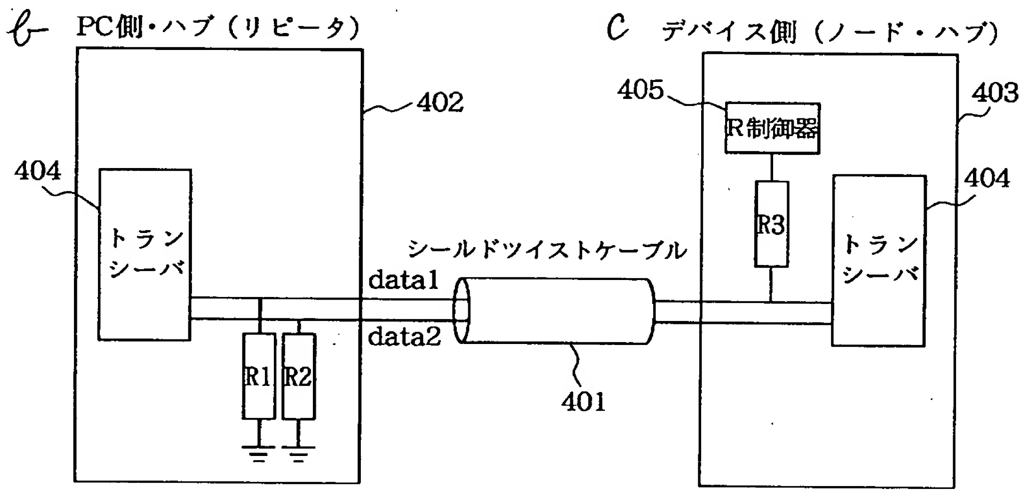
【図５】

Pin #	Pin assignments	Source
1	nStrobe	Host
2 − 9	Data	Bi − Di
10	nAck	Peripheral
11	Busy	Peripheral
12	PError	Peripheral
13	Select	Peripheral
14	nAutoFeed	Host
15	nFault	Peripheral
16	nInit	Host
17	nSelectIn	Host
18 − 25	Signal Ground	

【図６】


プリンタ60
印刷機構
ファームウエア
SIE (Serial I/F Engine)
HUB
55
56
57
58
Downstream Connector
Upstream Connector
USB Cable
ホスト50
ステータスモニタ
APIレイア
ドライバレイア
ホストコントローラ
51
52
53
54

【図７】


b
PC側・ハブ（リピータ）
c
デバイス側（ノード・ハブ）
402
403
404
405
R制御器
トランシーバ
R3
シールドツイストケーブル
data1
data2
R1
R2
401

【図８】

図8－1

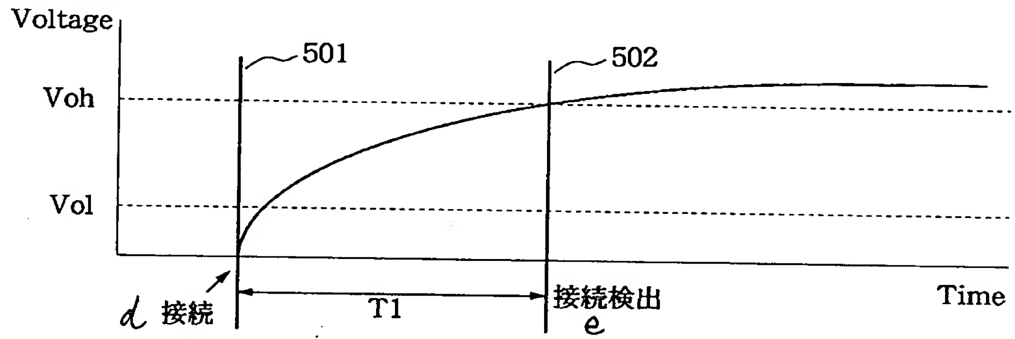
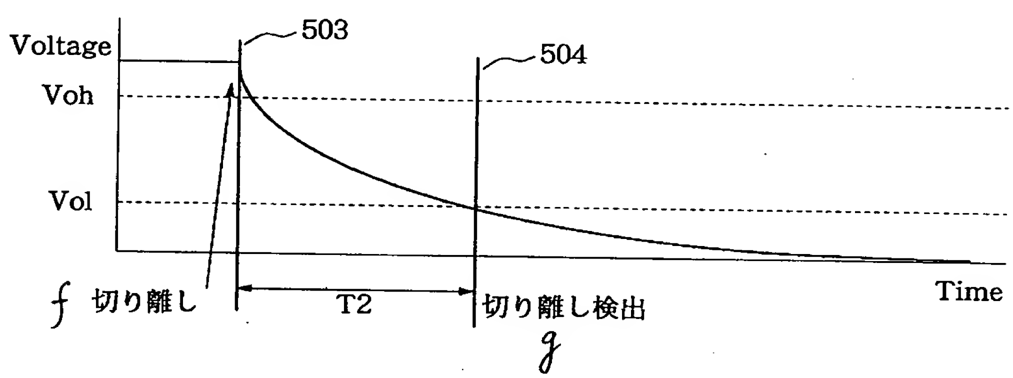
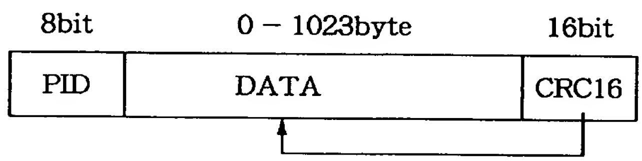

Voltage
501
502
Voh
Vol
d 接続
T1
接続検出
e
Time

図8－2


Voltage
503
504
Voh
Vol
f 切り離し
T2
切り離し検出
g
Time

【図９】



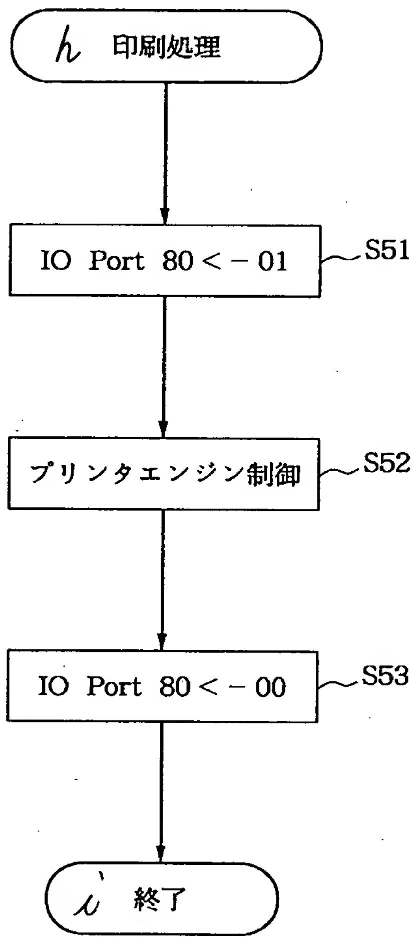
【図１０】

8bit	16bit
PID	CRC16

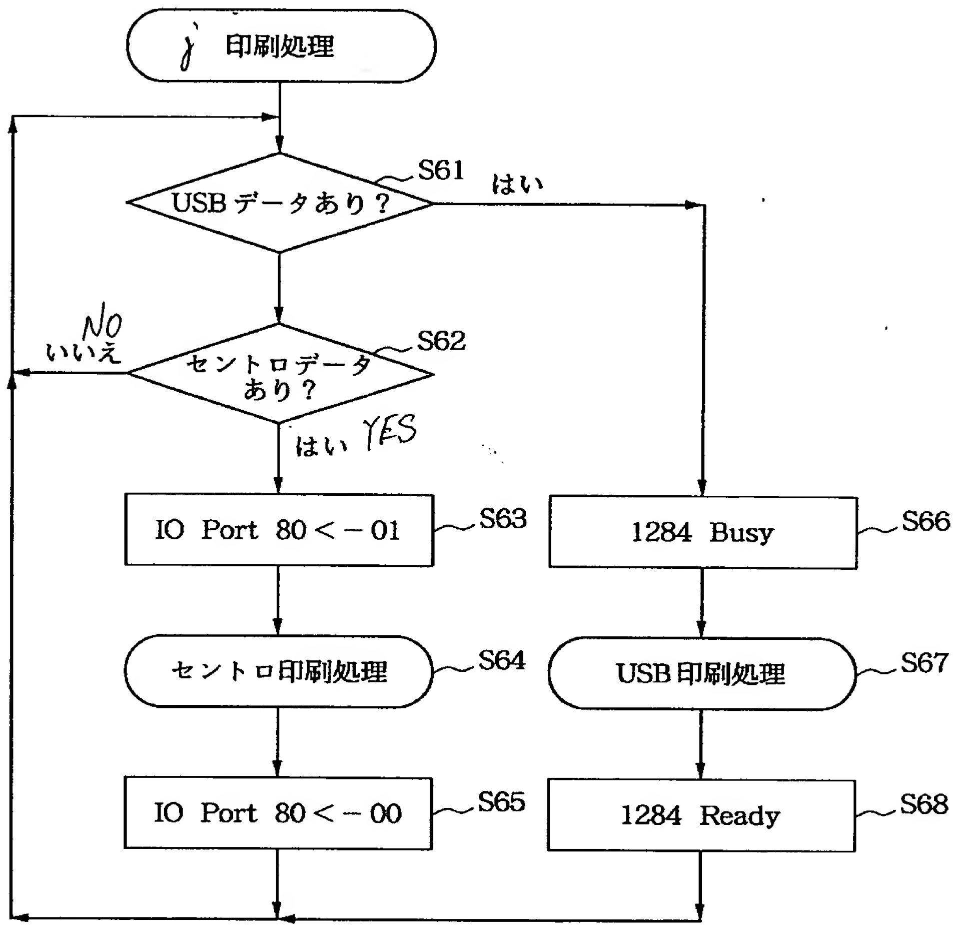
【図１１】

1byte
Nak

【図１２】



【図１３】


印刷処理
USBデータあり？
S61
はい
セントロデータ
あり？
S62
NO
いいえ
はい YES
IO Port 80＜－01
S63
セントロ印刷処理
S64
IO Port 80＜－00
S65
1284 Busy
S66
USB印刷処理
S67
1284 Ready
S68

[Name of Document] ABSTRACT

[Abstract]

[Object] To provide a peripheral apparatus that does not deteriorate the performance of a connected host.

[Solving Means] A peripheral apparatus includes connecting means (700) for connecting to a host computer; first means (600) for, in response to a data reading request issued from the host computer, notifying the connecting means of response data when the response data is ready and notifying the connecting means that a response cannot be made when the response data is not ready; second means (900) for notifying the connecting means that no status change occurs in the peripheral apparatus in response to the data reading request issued from the host computer; and switching means (1000) for switching the first means and the second means in accordance with a status of the peripheral apparatus.

[Selected Figure] Fig. 1

Applicant's Information

Identification No.		[000001007]
1.	Date of Change	August 30, 1990
	[Reason for Change]	New Registration
	Address	30-2, 3-chome, Shimomaruko, Ohta-ku, Tokyo
	Name	CANON KABUSHIKI KAISHA